

Movie Recommender System

Team Members:

Soheil Norouzi

Reza Yazdani

Professor:

Mr.Tourani

Course:

Computational Intelligence

What is a Recommendation System?

A recommender system, or a recommendation system (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications.

Recommender systems are utilized in a variety of areas and are most commonly recognized as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating. Recommender systems have also been developed to explore research articles and experts, collaborators, and financial services.

Overview

Recommender systems usually make use of either or both **collaborative filtering** and **content-based filtering** (also known as the personality-based approach), as well as other systems such as knowledge-based systems. Collaborative filtering approaches build a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. Content-based filtering approaches utilize a series of discrete, pre-tagged characteristics of an item in order to recommend additional items with similar properties. Current recommender systems typically combine one or more approaches into a hybrid system.

The differences between collaborative and content-based filtering can be demonstrated by comparing two early music recommender systems Last.fm and Pandora Radio.

- Last.fm creates a "station" of recommended songs by observing what bands and individual tracks the user has listened to on a regular basis and comparing those against the listening behavior of other users. Last.fm will play tracks that do not appear in the user's library, but are often played by other users with similar interests. As this approach leverages the behavior of users, it is an example of a collaborative filtering technique.
- Pandora uses the properties of a song or artist (a subset of the 400 attributes provided by the Music Genome Project) to seed a "station" that plays music with similar properties. User feedback is used to refine the station's results, deemphasizing certain attributes when a user "dislikes" a particular song and emphasizing other attributes when a user "likes" a song. This is an example of a content-based approach.

Each type of system has its strengths and weaknesses. In the above example, Last.fm requires a large amount of information about a user to make accurate recommendations. This is an example of the cold start problem, and is common in collaborative filtering systems. Whereas Pandora needs very little information to start, it is far more limited in scope (for example, it can only make recommendations that are similar to the original seed).

Content-based filtering

One approach to the design of recommender systems that has wide use is **content-based filtering**. Content-based filtering methods are based on a description of the item and a profile of the user's preferences.

These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user.

Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on an item's features.

In this system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past, or is examining in the present. It does not rely on a user sign-in mechanism to generate this often temporary profile. In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research.

To create a user profile, the system mostly focuses on two types of information:

1. A model of the user's preference.
2. A history of the user's interaction with the recommender system.

Now here we have a brief explanation of our Movie Recommender System using Content-Based filtering approach:

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

sklearn.feature_extraction.text import CountVectorizer Convert a collection of text documents to a matrix of token counts

This implementation produces a sparse representation of the counts using `scipy.sparse.csr_matrix`.

sklearn.metrics.pairwise import cosine_similarity Compute cosine similarity between samples in X and Y.

```
import warnings; warnings.simplefilter('ignore')
```

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program.

```
df = pd.read_csv("/content/drive/My Drive/my notebook/movie_dataset.csv")
```

The process of loading data from a CSV file into a Pandas DataFrame is achieved using the “read_csv” function in Pandas.

More Specific info of the Project is available on following Link:

https://drive.google.com/file/d/13Lq_QDCLy-5raF4ZKT3TZMwqJRBSVu0T/view?usp=sharing

Here we go to see how does the system work:

For example we give some movie names in different genres to machine:

1) Saw(horror)

please type one of your favorite movies: saw

whiteout
killing them softly
the house of the devil
cure
the general's daughter
black mass
nerve
checkmate
truth or dare
trance
shadow of the vampire
the green mile
along came a spider
captive
heaven is for real
the relic
indignation
the number 23
hansel and gretel get baked
eraser
shooter
dark water
silent hill: revelation 3d
your highness
gossip
ella enchanted

2)Scarface(crime drama)

please type one of your favorite movies: scarface

raising cain
righteous kill
the godfather: part ii
cop land
the untouchables
exit wounds
pain & gain
limbo
the glimmer man
buffalo soldiers
kill the messenger
10th & wolf
the son of no one
sea of love
l.a. confidential
code of honor
donkey punch
the scarlet letter
the score
up close & personal
i am wrath
brooklyn's finest
the life of david gale
the godfather
amidst the devil's wings
a most violent year

3)Fury(War)

please type one of your favorite movies: fury

saving private ryan

unbroken

hart's war

the monuments men

little boy

saints and soldiers

city of life and death

a bridge too far

miracle at st. anna

red tails

the thin red line

lion of the desert

u-571

patton

platoon

the boy in the striped pyjamas

the train

the great raid

catch-22

lawrence of arabia

windtalkers

tea with mussolini

flame & citron

all the queen's men

1941

war horse